Global Approximation Using Adaptive Regressive Polynomial Response Surfaces

Xiaoling Luo
Computer and Information Engineering College, Inner Mongolia Agriculture University
Hohhot, Inner Mongolia, China Country

luoxl@imau.edu.cndesired

***Abstract*-Global approximation substitute for the original model is also called response surface (RS), surrogate or meta-model. The key aspects should be considered when using the RS approximation method: the accuracy of the RS, the number of original model evaluations, the time consuming when updating the RS, the memory space occupied by the RS, and the speed of evaluation for a given point using the RS. This paper analyzes the drawbacks of the existing response surface methods, and then proposes an adaptive regressive polynomial response surfaces method using quadratic functions with domain decomposition. The test cases and applications effectively support the proposed method of this paper. As it is shown, the proposed method is far more effective and accurate.**

***Keywords*-global approximation, polynomial response surface, domain decomposition, re-compose, hybrid electric vehicle**

I. INTRODUCTION

It is very applicable and useful to substitute a complex nonlinear function or simulation model (now we call it original model) approximately with another simple model under some special occasions such as parametric experiments, sensibility analysis or design optimization for the original model. As a typical example, the approximation for an original finite element analysis model can greatly speed up the process of parametric optimization for the model. In the system simulation field, when an original model (such as an engine or a powertrain model) is used as a component in a top-level system (such as a vehicle system), the relationship from the inputs to the outputs of the component can be approximated by a simple model while simulating the whole system. Look-up table and interpolation is a method to solve this type of problems, but it is just suitable for the lower-dimension problems, i.e., with few input variables. More important, it is difficult to build a look-up table with the appropriate grids among the global domain of the inputs variables.

A. Kriging

Kriging technique was named after the South African mining engineer D.G. Krige [2] and was introduced to computer experiments by Sacks et al [3]. Kriging model estimates the value of an original model as a combination of a known function and departures which are realized by a stochastic process from a Bayesian perspective [4]. A conventional Kriging model interpolates the sampling points, which is an important characteristic for design and analysis of computer experiments. Kriging is also a flexible technique since different instances can be generated by choosing different pairs of regression and correlation functions [5]. In addition, Kriging is suitable for both high order functions and low dimensional problems, and it can approximate the original model smoothly with relatively fewer sampling points.

B. Radial basis functions

RBF technique was firstly introduced by Hardy R. L. to fit irregular topographic contours of geographical data [6]. A RBF model is expressed as the linear sum of a series of basic functions about the sampling points [7]. According to the existing researches, RBF performs well for high dimensional and high-order nonlinear problems, and the accuracy of RBF is between PRS and Kriging, while RBF model is easier to be constructed than Kriging model [8]. Radial basis neural network (RBNN) is a combination of RBS and the artificial neural network (ANN) which uses radial basis functions as transfer functions of ANN [9]. Compared with the standard feed-forward or back-propagation networks, RBNN may require more neurons, but generally it can be generated in a fraction of the time that takes to train standard networks.

C. Polynomial response surface

PRS approximation is another well established meta-modeling technique, which is first developed and described by Box and Wilson [10]. In PRS modeling, a second-order polynomial function is commonly used to approximate the original model [11]. The coefficients of the polynomial functions can be obtained by least squares and the fitting is unbiased and have minimum variance. PRS has another characteristic that it is possible to identify the significance of different design factors directly from the coefficients in the normalized regression model [12].

D. Support vector regression

SVR, a particular implementation of support vector machines [13], is introduced as an alternative technique for approximating complex engineering analyses [14, 15]. The SVR calculates the optimal hyperplane for the training data. The aim of SVR is to find a fitting function that has at most a deviation of magnitude from each of the training data. Mathematically, SVR is expressed as a constant plus the linear sum of a series of kernel functions about the training points. The parameters are obtained from a fitting process during which SVR minimizes an upper bound on the expected risk using alternative loss functions. According to S. M. Clarke et al [16], SVR had the best overall performance for the test bed of 26 engineering analysis functions in comparison to other approximating techniques. SVR also gives a robust approximation, providing a good compromise between prediction accuracy and robustness.

E. Multivariate adaptive regression spline

MARS is a non-parametric regression technique that constructs the relationship from inputs to outputs variables through spline theory [17]. The input space is divided into regions containing their own regression equation which contains knots vectors and control points. MARS is suitable for problems with high input dimensions and has been applied in stochastic dynamic programming [18] and global optimization [19]. S. Richardson also implemented a multivariate adaptive regression B-spline algorithm (BMARS) for solving a class of nonlinear optimal feedback control problems [20].

F. Analysis

The key aspects should be considered when using the RS approximation method to solve a real engineering analysis problem: the accuracy of the RS, the number of original model evaluations, time consuming to construct and refine the RS, the memory space occupied by the RS, and the speed of evaluation for a given point using the RS. Needless to say, the accuracy of the RS is the basic requirement of the approximation. The number of the evaluations for the original model should be limited because each evaluation may be expensive. The approximation procedure is a refining iteration, so time spent during each iteration of the refining step should be limited. Next, the more effective the technique, the smaller the memory space occupied by RS, and the faster the evaluation for a given point using the RS.

For Kriging technique, it is suitable for high order functions and low dimensional problems, but it is less efficient for low order functions and high dimensional problems [21]. And because it belongs to an interpolation method, it is high sensitive to the noise data. Through testing DACE code [22], it is not difficult to find that if the initial values of correlation function parameters are set un-suitably, the result will be nothing than what we expect. In addition, different selections of the regression function and correlation function will result in different instances, but for a particular problem how to choose a suitable pair?

For RBF technique, it performs well for high dimension and high-order nonlinear problems, but it has trouble in dealing linear problems, and it is less efficient for low dimensional problems due to its slow convergence [23]. For RBNN, its crucial drawback for some applications is the need of many training points [1].

For PRS technique, it is particularly suitable for lower order functions and low-to-medium dimensional problems near a local area. Although some adaptive response surface methods are proposed [24,25], it performs poorly for high order functions and high nonlinear models upon a global large domain.

For SVR technique, because its fitting process needs a optimal search for the upper bound on the expected risk, the constructing and refining procedure must be low efficient.

All these approximation techniques needs sampling points to construct meta-model. Classical sampling methods about design of experiments such as central composite design, fractional/full factorial design or D-optimal design cannot meet the accurate requirement when the domain of the inputs variables is large. Space filling sampling such as grid design, random sampling or Latin hypercube sampling can not provide enough but not too many points adaptively according to the nonlinear characteristic of the original model. This is one reason why we do not attempt to construct one single response surface upon a global large domain. Under most occasions, we need to split the domain into some smaller cells.

MARS uses the decomposition idea to construct an interpolation response surface upon a global domain. According to spline theory, constructing a spline surface needs reverse calculation procedure which is time consuming when the dimension is high. According to [17], MARS algorithm includes calculation for optimal lack-of-fit procedure which is much time-expensive.

Armin Iske and Jeremy Levesley [26] studied a method for multilevel scattered data approximation by using compactly supported radial basis functions with adaptive domain decomposition. The numerical examples show that the new method achieves an improvement on the approximation quality of previous well-established multilevel interpolation schemes. Other researchers studied the adaptive sampling and global approximation for RBF [27-28]. But with a large domain, high dimension and then many sampling points, the RBF refining process is inevitably low efficient.

D. Busby et al [29] proposed a hierarchical nonlinear approximation method for experimental design and statistical data fitting. The iterative method combines at any refinement step the selection of suitable evaluation points with Kriging for statistical data analysis. Recent improvements on Krging technique can be found in [30-31]. Like the domain decomposition with RBF, the refining process is also low efficient, the memory the metamodel occupies is large, and then evaluation for given points is slow.

For PRS, D. Shahsavani and A.Grimvall [32] researched an adaptive design and interpolation technique for extracting highly nonlinear response surfaces from deterministic models. "A sequential design algorithm for cuboid domains is initiated by selecting an extended corner/centre point design for the entire domain, then updated by decomposing this domain into disjoint cuboids and taking the corners and centre of these cuboids as new design points". As we know, numbers of corner and center points for a p-dimension cubic is

$$n_1 = 1 + 2^p$$

while the quadratic polynomial items of the p-dimension is

$$n_2 = 1 + 2p + C_p^2$$

When p≤3, n1<n2, interpolation can be done upon the cubic cell through adding some additional points. But if p>3; n1>n2, it is impossible to interpolate the cell with a quadratic polynomial function. So I think this method cannot meet multi-dimension problems. In addition, the testing or sampling data are lost during the refining process.

II. PROPOSED METHOD

Due to the drawbacks of the existing response surface methods with a global large domain, this paper proposes an adaptive regressive polynomial response surfaces method using

quadratic functions with domain decomposition. During the iteration for updating RS set through Latin hypercube testing, the roughest cells are selected and split along the roughest dimension direction to decompose the domain. After the RS set is accurate enough according to some criteria, the domain combination process is executed to unite the adjacent cells so that the number of the response surfaces in the RS set can be eliminated while keeping the given accuracy. During the whole refining process, all the sampling data which may be obtained through expensive calculation are utilized. Moreover, the paper also proposes an effective multi-dimension cells sorting algorithm to evaluate a given point using the generated RS set.

A. *Mathematical Foundations of the Method*

For an original model with k outputs and p inputs or dimensions, described as:

$$\boldsymbol{y} = f(\boldsymbol{x}),$$

where y: $\Omega \subset R^k$, x: $\Omega \subset R^p$; and f: $R^k \to R^p$.

We can handle the y (k outputs) into k functions set:

y = [y1(x), y2(x), …, yk(x)]

Then for each of the ith scalar output (1≤i≤k) yi, we will construct its own response surface. As it is known, the quadratic response surface can be expressed as:

$$\hat{y}(\boldsymbol{x}) = \beta_0 + \sum_{i=1}^{p} \beta_{ii} x_i^2 + \sum_{i=1}^{p} \sum_{1 \le i < j} \beta_{ij} x_i x_j \tag{1}$$

whereβare coefficients determined through least squares regression.

For the global domain, it is impossible to construct a simple polynomial quadratic response surface upon the whole scope. So we adopt the recursive partitioning procedure during which the worst cells (which maybe have the worst accuracy) will be selected through adding some sampling points and then be decomposed with two binary equal cells along the roughest dimension at each step.

B. *The Worst Cells*

Theoretically, the roughness of a cubic cell C can be evaluated according to the coefficients of the quadratic PRS and each side-length of C [32]:

$$\hat{R}(C) = V(C)\left[\sum_{j=1}^{p} (2\beta_{jj} l_j^2)^2 + 2\sum_{j<k} (\beta_{jk} l_j l_k)^2\right]$$

where V(C) denotes the volume of C, lj indicates the length of the j th side of C.

However, in order to take the sampling data into account and to be compatible with other techniques or high order polynomial functions, we proposed a container-box method. As we suppose, the worst cell is the one which has the largest container box of the sampling points. As an example with one dimension problem in Fig.1(a), the volume of the container box equals distance a times height h, where a is the length of the vector a which is from the left-down corner point to the right-up point of the cell; h is the height of two line which are parallel to vector a.

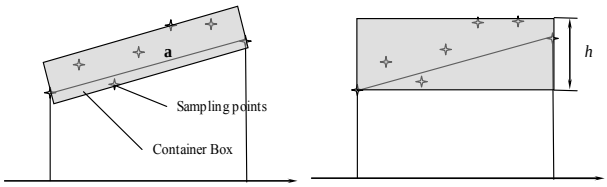


Fig.1 Roughness Calculation of a Cell

But for the multi-dimensional problem, it is difficult to calculate the volume of the container box. So we can approximate the value with a simple expression like Fig.1b) shows:

$$R(C) = hV(C) \tag{2}$$

where $h = \max_{1 \le i \le N}(y_i) - \min_{1 \le i \le N}(y_i)$, and yi is the response value of ith point of the N sampling points.

C. *The Splitting Dimension*

Similarly, the roughest dimension can be determined by the splitting direction dimension criterion which is calculated from the coefficients of quadratic PRS [32],

$$\hat{S}(j) = (2\beta_{jj} l_j^2)^2 + \sum_{1 \le k < j} (\beta_{kj} l_k l_j)^2 + \sum_{p \ge k > j} (\beta_{jk} l_j l_k)^2$$

and the splitting dimension js is:

$$j_s = \arg\max_{1 \le j \le p}(\hat{S}(j))$$

In the same way, by utilizing the sampling data, the criterion can be simplified as

$$S(j) = \max_{1 \le i \le N_j}(h_i) \tag{3}$$

hi is the height of the ith sample point along j direction dimension. As the Fig2 shows, hi can be calculated:

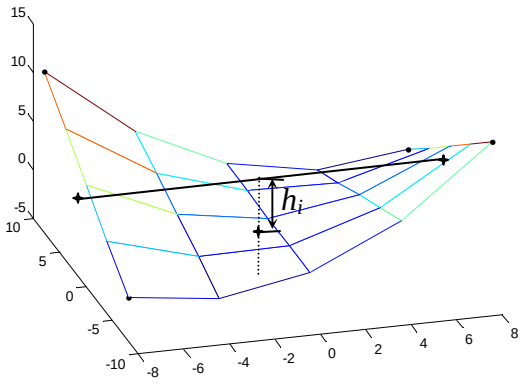


Fig.2 Approximation of the Splitting Dimension

$$h_i = |y_i - y_i'|$$

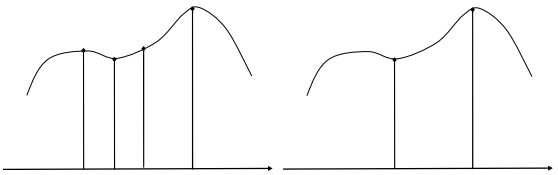
yi is the response value of the ith sampling point upon the cell, yi' is the interpolation value of the two values of the two side points of the cell which have the same coordinates except j dimension and can be evaluated by the PRS function.

Then the splitting dimension js is

$$j_s = \arg\max_{1 \le j \le p}(S(j)) \tag{4}$$

D. Re-composing the RS Set

After all the cells are refined and accurate enough, in order to eliminate the number of cells it is probable to unite some adjacent cells into one cell while keeping its accuracy. As the Fig3.a) shows, the domain is divided into 5 cells after recursive partitioning, apparently, we can re-compose these cells into 3 cells like Fig3.b) shows.



(a) (b)

Fig.3 Cells Recomposing

Mathematically, two cells can be re-composed into one cell if the following two conditions are met:

1) Domain adjacent constraint:

We assume that XL, XU are left-down (with minimum values of all dimensions), right-top (with maximum values of all dimensions) points of a cell's cubic respectively. Then if two cells C1 and C2 can be united, they should have the relations:

$\forall$ j $\in$ {1, 2, …, p}

XL1(j) = XU2(j) or XL2(j) = XU1(j)

XL1(i) = XL2(i) and XU1(i) = XU2(i) (i ≠ j)

2) Accuracy constraint

We assume that fnew is the regressive quadratic PRS function using all the N sampling points upon the two cells; pti is the ith sampling points. Then the RSME (root square mean error) of the new RS should less-equal than a threshold RSME0:

$$RSME = \sqrt{\left(\sum_{i=1}^{N}(y_i - y_i')^2\right) \Big/ N} \le RSME0$$

Where yi is the response value of pti;

$$y_i' = f_{new}(pt_i)$$

E. Sorting the RS Set

Although using the re-composing procedure, the cells of the generated RS set may be hundreds or thousands when the domain is large and the model is high non-linear. Evaluating a given point x using the RS set should position the cell (XL, XU) it belongs to, which means,

XL ≤ x ≤ XU

If the RS set has a large number of cells, this positioning process is also time-consuming. So we proposed a cells sorting and binary search method which can speed up the positioning process.

In order to sort the cells, we can define '≥' operation,

Definition 2: for two cells C1, C2,

C1 ≥ C2

While p1 (the left-down corner point of the cell C1) is greater and equal than p2 (that of C2),

p1 ≥ p2

if the first unequal coordinate value p1(j) > p2(j), where 1 ≤ j ≤ p.

F. Generating of RS Set through Step-By-Step Refining

So we start with the whole domain Ω with the initial RS0, an object of RS structure which occupies the domain cube, coefficients, sampling points including corners and center points, testing points.

Definition 1: A cell is a basic domain including a cube (or scope) and a response surface.

Here we give the RS object description at first. Data structure of RS has the following fields:

Cube: records the scope of the RS with XL, XU;

B: records the coefficients of RS function;

R; records the residue of regression;

FC: records the fitting counts of testing passed

SD: records the sampling data (points and response values) which are used to regress the RS function

TD: records the testing data which are not used for regression

G. Algorithm Steps

The step-by-step refining procedure is as following,

Step1: construct an initial RS0 upon theΩ, and then execute the first recursive partitioning for RS0, and generate the initial RS set RSS = RSS0;

Step2: if some stopping criterion reaches, then exit; else go to next step;

Step3: get the probable worst cells CS from the RSS according to the calculation from (2); and divided by the RS's fitting counts powers of p;

$$R(C) = R(C) \Big/ p^{RS.FC} \tag{5}$$

where C is one of CS, RS is one element of RSS.

Step4: for each cell C in CS, adopt LHC sampling on C with p points, run the original model to obtain the pair of response value-points, <vals, pts>, and add the pairs into the TPS and TV of the RS; then determine whether the pairs of sampling points and response values satisfy the RS function of C: if it is true, increase the fitting counts of RS upon C; else record the RS into the real worst cells RCS;

Step5: for each cell in RCS, execute the recursive partitioning; obtain subsets SS of RCS;

Step6: update the RSS with SS.

Now, some additional explains should be given in the above steps.

H. Recursive Domain Partitioning

If a RS is not accurate, recursive domain partitioning should be executed. The function is as following,

function BinarySplit

Input: rs – response surface

Output: RSset – response surface set

Step1: split the domainΩ of rs into two equal ones:Ω1 andΩ2 along the roughest dimension according to (3) and (4);

Step2: distribute the sampling data and testing data of rs into the two sun-domains according to their coordinates;

Step3: get the corners and centers of the two sub-domains, calculates their response values through running the original model if some of them are not calculated;

Step4: call regression procedure with key data (corners and center), sampling and testing data to generate two RSs of the two sub-domains respectively; update the sampling data with all the above data and reset the testing data to null;

Step5: for each one RS of the two RSs, if the residue error is not accurate, call BinarySplit function recursively;

Step6: record all the RSs into RSset.

I. Regression or Interpolation

As mentioned above, the center and the corners are key points of a domain. We should calculate and take them into account for RS regression because the key data take a big influence on the RS accuracy.

So for a domain with p dimensions, if $p \leq 3$ and there are no enough other sampling points upon it, the regression procedure executes interpolation automatically; otherwise (if $p > 3$ or there are enough sampling points upon the domain for regression) the RS will be regressed.

J. Stopping Criterion

We adopt 4 criteria to stop the refining procedure of RS set as following:

i) Number of iterations.

ii) Number of runs of the original model.

iii) Side-lengths of all the RS cells reaching to a threshold, which means that if all the cells are small enough, the iteration will stop.

iv) Fitting counts of all the RS in RS set. When they are greater than a given integer M (M=2), the iteration stops. This means that not only the RSs in the RS set satisfy for the sampling data but also it passed M random sampling tests. So we can assume that all the RSs are accurate enough.

K. About the probable worst cells

In step3, the probable worst cells are selected among the RS set. The roughness of a cell can be evaluated according to (2), but this is not the unique criterion. Because if the RS has passed FC tests, this RS may be accurate. So we adopt the expression (5) for roughness evaluation according to lots of numerical experiments.

Even if the worst cells are chosen from (5), still the cells need successive sampling tests. Only if the probable worst cell cannot pass the sampling test, they will be assumed to be a real worst cell.

L. Testing the RS

Before splitting the probable worst cells, we should determine whether the cell is a real worst one by executing a sampling test. Here, we adopt the most commonly-used random sampling technique, i.e., Latin hyper-cubic sampling. Too many or too few sampling points are both unsuitable: too many points may be time consuming when running the original model; too little points may not achieve the testing object. Here, we set the number of sampling points to p, the dimensions of the domain.

How to determine whether a cell has passed the sampling test successfully? We adopt four criteria: absolute error, relevant error, side-lengths and RSME of the sampling data.

M. Updating the RS set

After executing the recursive partitioning upon a cell, a new subset of RSs will be generated. The updating process is to substitute the cell with the new subset.

During the binary splitting procedure, at each calling one cell will be divided into two cells. So after each splitting, a subset including two RSs will be generated and the original one will be substituted with the subset of two RSs.

N. Re-composing of RS Set

The number of the result RS set for a domain approximation directly influences the memory and evaluation with RS set. Apparently, the less the number of the RS set, the smaller the memory and the faster the evaluation with the RS set. So after decomposing the domain it is necessary to recompose the adjacent cells while keeping the accuracy.

In order to speed up the recomposing procedure, sorting process should be executed in advance. This sorting process algorithm is based on the idea in 2.4. The re-composing procedure upon the sorted RS set is expressed as following:

Step1: for each RS in the current RS set RSS

Step2: generate two testing points which belongs to the adjacent cells and extend just a little displace δ along the positive direction of all dimensions, like Fig4 shows.

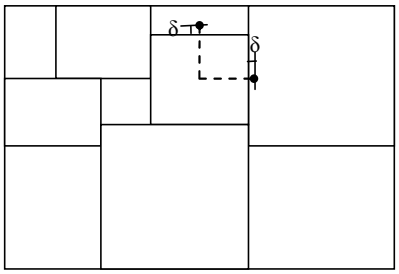


Fig.4 Testing Points to Obtain the Adjacent Cells

Step3: get the two cells at most which contains the two testing points.

Step4: remove the cell(s) which has no co-edge with the RS, RSsetA;

Step5: for the each cell rsi in RSsetA, if rsi and RSk can be united according to 2.3, regress a new RS using the sampling and testing data upon the two cells; reset the sampling data of the united cell and replace the original two cells with the new one; update the RSS.

Step6: if uniting process is called, then go to step1; else exit.

In the step3, we can determine whether a given point is in the RS cell according to the coordinates comparison. But because there may be many RS in the RS set, so we adopt a so-called quasi binary search to position the right RS quickly in the next part.

O. Evaluating using RS Set

Evaluating using the RS set is the main and direct object of approximation. As we know, it is very efficient to evaluate a given point for a quadratic PRS using the function (1). But if the domain is too large and the response surface is high dimension, and the number of the RSs in the resulting RS set may be very large, then the positioning for the given point may be time-consuming if executing the one-by-one search process.

Based on the sorted RS set, we can adopt the following quasi binary search to quickly position the particular RS for the given point.

Algorithm 1 (Binary-Positioning)

midlen ← length(RSset) / 2;

start ← 1 + midlen;

while loop

RSMid ← RSset[start + midlen];

if pt is in RSMid

RSMid is found and return;

elseif pt is after RSMid

start ← start + midlen;

midlen ← midlen / 2;

else

midlen ← midlen / 2;

end if

end while loop.

end algorithm

During the above procedure, we define "a pt is after a cell C" means that pt follows the left-corner of C (the point with least value of all dimensions in a cell domain) which is similar to definition 2.

The above algorithm is a typical binary search procedure. However, the right RS maybe can not found. As an example like Fig.5 shows, the point pt is after the RS cell 8, but the real RS set we want is RS cell 5, which is before 8.

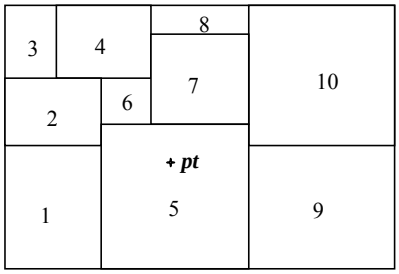


Fig.5 Position the Cell which Contains a Given Point

The main reason to cause such a problem is that: if a point pt is before a RS cell 10, the cell A containing pt is absolutely before 10; but if pt is after the cell B, A is not absolutely after B. Now, we define an "absolute after" function from definition 2,

Definition 3: for a point pt and a cell C,

pt >= C,

when pt(j) >= C.XL(j), where $1 \le j \le p$.

So we modify the binary search algorithm as following:

Algorithm 2 (Quasi_Binary_Positioning)

midlen ← length(RSset) / 2;

start ← 1 + midlen;

nFrom ← 1;

while loop

RSMid ← RSset[start + midlen];

if (start+midlen>len) or (midlen<=1)

no results found and break;

if pt is in RSMid

RSMid is found and return;

elseif pt is before RSMid

nTo ← start + midlen;

midlen ← midlen / 2;

else

if pt is absolutely after RSMid

nFrom ← start + midlen;

end if absolutely after

start ← start + midlen;

midlen ← midlen / 2;

end if

end of while loop.

if not found yet

Search from nFrom to nTo cells one-by-one;

end if

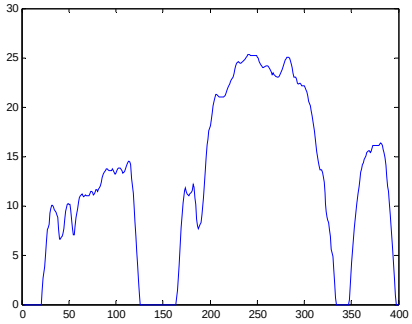
end algorithm

From above algorithm, we can see that the result cell can be found during the binary search process on some occasions. And even if no results are found during the binary search, the search scope can be eliminated, that means after the binary search, the next one-by-one search can be executed from the nFrom to nTo.

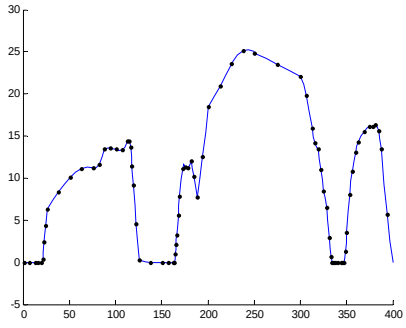
III. EXPERIMENTS

A. *Approximation for 1-D Data*

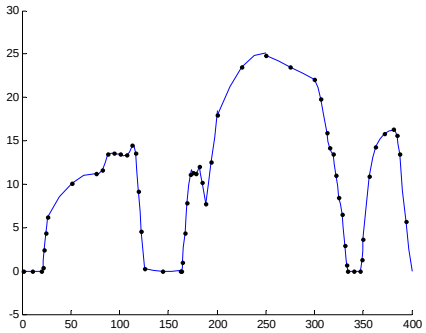
As fig.6 shows, the 1-D curve is abstracted from the UDDS driving cycle which stands for Urban Dynamometer Driving Schedule, and refers to an United States Environmental Protection Agency mandated dynamometer test that represents city driving conditions which is used for light duty vehicle testing.



(a) 1-D Driving Cycle – UDDS Curve



(b) Approximate Result after Decomposing



(c) Approximate Result after Re-Uniting

Fig.6 Approximation for 1-D Problem

Generally, the driving cycles are recorded using the look-up table which includes the velocity values at each second. As for Fig.6, there are 400 data points to describe the driving cycle.

Now, we approximate the drive cycle using the adaptive response surface domain decomposition, we can obtain the approximate RS set (39 cells) in Fig6.b); and after executing the re-composing algorithm setting the RSME = 0.1, the result RS set (29 cells) is shown in Fig6.c). Now, the number of the data points which can approximately record the original driving cycle is 29×2+1=59, which is much less than the original number 400.

Note that the result may be different with different runs because of the LHC random sampling.

B. *Approximation for a Powertrain Model*

The key object of the control optimization for a hybrid electric vehicle is to improve the fuel economy of the powertrain. Determinative dynamic program (DDP) is the commonly-used method for control optimization. During the DDP process, conventionally the powertrain simulation model should be called and simulated to calculate some outputs with the inputs (control variables and state variables) for a given time step (1 second). Inevitably this process is quite calculation-expensive.

Now we will approximate the Prius powertrain model which has one internal-combustion engine and two motors like Fig.7 shows.

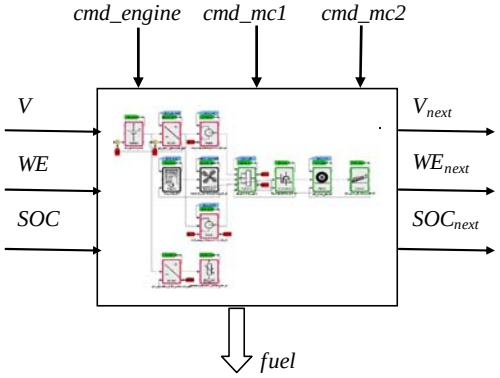


Fig.7 Powertrain Control Model of Prius

Form the above figure, we can see that there are 4 outputs: fuel – the fuel consumed, Vnext – the vehicle speed of the next time step, WEnext – the engine speed of the next time step, SOCnext – the state of charge of battery of the next time step. And then there are 6 inputs: three control variables (cmd_engine - throttle of engine, command of the two motors: cmd_mc1 and cmd_mc2) and three state variables (V – the current vehicle speed, WE – the current engine speed, SOC – the current SOC of the battery.

For the four outputs, four corresponding response surface sets are generated respectively with the same six input variables. If setting the stopping criterion with the maximum iterations N = 100, we can get four different RS sets of the outputs: fuel, Vnext, WEnext and SOCnext.

TABLE IAPPROXIMATION RESULT OF POWERTRAIN CONTROL MODEL OF PRIUS

	fuel	**Vnext**	**WEnext**	**SOCnext**
Cells	1293	4323	3234	2344
RSME	1E-6	0.10	10	0.001

IV. CONCLUSIONS

The proposed approximate method in this paper meets the five requirements: the accuracy can be controlled according to the set-up; the number of original model evaluations is eliminated because we adopt an adaptive recursive partitioning process; it is very efficient when updating the RS set because all the RSs are quadratic polynomial functions; the memory space occupied by the RS is also quite small because we adopt QPRS and execute the re-composing process; and the speed of evaluation for a given point using the RS set is also efficient because of using QPRS and quasi binary searching.

There are some other characteristics: it tolerates few bizarre sampling data because we use regression instead of interpolation; it utilizes all the sampling data during the entire algorithm, which are obtained through expensive calculations; it provides several criteria to stop the recursive partitioning and the whole algorithm; The decomposition method can also be used in other approximation techniques such as Kriging or RBF.

However there still exist some drawbacks of the method, one of them is that it is 'discontinuous at the sub-region boundaries' because of regression, but even imposing continuous conditions at the key points, it is impossible to guarantee that all the points on the edge are continuous. Another one is that how to handle multiple outputs using all the sampling data when generating each RS set in parallel.

ACKNOWLEDGMENT

This work was supported by the NDPYTD 2010-9.

REFERENCES

[1] SURROGATES TOOLBOX, http://fchegury.googlepages.com

[2] G. Matheron, Principles of geostatistics, Economic Geology, 58 (1963), pp. 1246–1266.S. Zhang, C. Zhu, J. K. O. Sin, and P. K. T. Mok, "A novel ultrathin elevated channel low-temperature poly-Si TFT," IEEE Electron Device Lett., vol. 20, pp. 569–571, Nov. 1999

[3] Sacks, J., Welch, W. J., Mitchell, T.J., and Wynn, H. P., 1989, "Design and Analysis of Computer Experiments", Statistical Science,, Vol.4(4), pp.409-423.

[4] Cresssie, N., 1988, "Spatial Prediction and Ordinary Kriging", Math. Geol., Vol.20(4), pp. 405–421.

[5] S. N. Lophaven, H. B. Nielsen, and J. Søndergaard, "DACE - A MATLAB Kriging Toolbox," Technical Report IMM-TR-2002-12, Informatics and Mathematical Modelling, Technical University of Denmark, 2002.

[6] Hardy, R. L., 1971, "Multiquadratic Equations of Topography and Other Irregular Surfaces", J.Geophus Res., Vol.76, pp.1905-1915.

[7] M. D. Buhmann, Radial Basis Functions, Cambridge University Press, Cambridge, UK, 2003.

[8] Martin, J. D., and Simpson, T. W., 2003, "A Study on the Use of Kriging Models to Approximate Deterministic Computer Models", Proceedings of 2003 ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Chicago, Illinois USA.

[9] Smith, M., 1993, "Neural Networks for Statistical Modeling," Von Nostrand Reinhold.

[10] Box, G. E. P., and Wilson, K. B., 1951, "On the Experimental Attainment of Optimum Conditions", Journal of the Roal Statistical Society. Series B(Methodological), Vol.13(1), pp.1-45.

[11] G. E. P. Box, W. G. Hunter, and J. S. Hunter, Statistics for experimenters, John Wiley & Sons, New York, 1978.

[12] Myers, R. H., and Montgomery, D. C., 2002, "Response Surface Methodology: Process and Product Optimization Using Designed Experiments", John Wiley& Sons, INC., Toronto.

[13] S. R. Gunn, "Support Vector Machines for Classification and Regression," Technical Report, Image Speech and Intelligent Systems Research Group, University of Southampton, UK, 1997.

[14] Vapnik, V., Golowich, S., and Smola, A., 1997, "Support Vector Method for Function Approximation, Regression Estimation, and Signal Processing," Advance in Neural Information Processing Systems, MIT Press, Cambridge, MA, pp. 281–287.

[15] Prakasvudhisarn, C., Trafalis, T. B., and Raman, S., 2003, "Support Vector Regression for Determination of Minimum Zone," ASME J. Manuf. Sci. Eng., 125_4_, pp. 736–739.

[16] S. M. Clarke, J. H. Griebsch, and T. W. Simpson, "Analysis of Support Vector Regression for Approximation of Complex Engineering Analyses," Journal of Mechanical Design, Vol. 127, pp. 1077-1087, 2005.

[17] Friedman, J. H., 1991, "Multivariate Adaptive Regression Splines", The Annals of Statistics, Vol.19(1), 27 pp.1-67.

[18] Chen, V. C. P., Ruppert, D., and Shoemaker, C. A., 1999, "Applying Experimental Design and Regression Splines to High-Dimensional Continuous-State Stochastic Dynamic Programming," Oper. Res., 47, pp. 38–53.

[19] Scott Crino, Donald E. Brown, Global Optimization With Multivariate Adaptive Regression Splines IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS, VOL. 37, NO. 2, APRIL 2007 pp333-340

[20] Steven Richardson, Song Wang, Les S. Jennings. A multivariate adaptive regression B-spline algorithm (BMARS) for solving a class of nonlinear optimal feedback control problems. Automatica 44 (2008) 1149 – 1155

[21] Gu, J., Li, G. Y., and Dong, Z., 2009, "Hybrid and Adaptive Metamodel Based Global Optimization," Proc. Proceedings of the ASME 2009 International Design Engineering Technical Conferences &Computers and Information in Engineering Conference, IDETC/CIE 2009, DETC2009-87121, August 30 - September 2, 2009, San Diego, California, USA.

[22] S.N. Lophaven, H.B. Nielsen, J. Sindergaard, Aspects of the Matlab Toolbox DACE. Report IMM-REP-2002-13, Informatics and Mathematical Modelling, DTU. (2002), 44 pages. Available as http://www.imm.dtu.dk/»hbn/publ/TR0213.ps

[23] Younis, A., Gu, J., Dong, Z., and Li, G., 2008, "Trends, Features, and Tests of Common and Recently Introduced Global Optimization

Methods," 12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference Victoria, British Columbia Canada, Paper No. AIAA-2008-5853.

[24] James R. Simpson , Drew Landman, Rupert Giroux et al, Adapting Second-order Response Surface Designs to Specific Need. QUALITY AND RELIABILITY ENGINEERING INTERNATIONAL. Qual. Reliab. Engng. Int. 2008; 24:331–349

[25] Xuan Son Nguyen, Alain Sellier Frédéric Duprat et al. Adaptive response surface method based on a double weightedregression technique. Probabilistic Engineering Mechanics 24 (2009) 135–143

[26] Armin Iske and Jeremy Levesley. Multilevel scattered data approximation by adaptive domain Decomposition. Numerical Algorithms (2005) 39: 187–198

[27] [Tobin A. Driscoll_, Alfa R.H. Heryudono. Adaptive residual subsampling methods for radial basis function interpolation and collocation problems. Computers and Mathematics with Applications 53 (2007) 927–939

[28] HONGBING FANG and MARK F. HORSTEMEYER. Global response approximation with radial basis functions. Engineering Optimization. Vol. 38, No. 4, June 2006, 407–424

[29] DANIEL BUSBY, CHRIS L. FARMER, AND ARMIN ISKE. HIERARCHICAL NONLINEAR APPROXIMATION FOR EXPERIMENTAL DESIGN AND STATISTICAL DATA FITTING. SIAM J. SCI. COMPUT. Vol. 29, No. 1, pp. 49–69

[30] Jay D. Martin. Computational Improvements to Estimating Kriging Metamodel Parameters. Journal of Mechanical Design. AUGUST 2009, Vol. 131 / 084501-1

[31] Jack P.C. Kleijnen. Kriging metamodeling in simulation: A review. European Journal of Operational Research 192 (2009) 707–716